General Architecture for Semantic Querying of Heterogeneous Data Sources*

Ivo Marinchev[1]

***Abstract* – In this paper we propose a general architecture for semantic querying of heterogeneous data sources. The general idea is to introduce semantic descriptions in the forms of base ontologies to the legacy data sources containing structured data. All base ontologies are mediated to a general ontology that describes the whole domain knowledge against which all end user applications are created. The advantages of this approach are twofold. First – all applications are written independent of the physical and logical representation of processed data. And second – at any time data sources can be added/removed to the application stack by additions/removal of only the corresponding transformation and data access mapping rules that are part of the data mediator layer.**

***Keywords* – Data integration, Ontology, Heterogeneous data sources, OWL, RDF, SPARQL.**

I. Introduction

In recent years the need to process data from heterogeneous data sources becomes widespread. It is the result of integration of many legacy databases that were developed for use in proprietary applications but later (sometimes many years after they were created) data integration becomes a central issue in many areas to facilitate the access and manipulation of highly distributed, heterogeneous and dynamic collection of information sources.

Integrating and querying data from heterogeneous sources is a hot research topic in database research field. The goal of data integration is to provide user a uniform access to multiple heterogeneous data sources. This problem is known in the literature as query rewriting and query answering using views, and has been studied very actively in the recent years [10]. However, with the use of ontology, these former research works are not applicable.

In this paper, an ontology-based approach for heterogeneous data source integration is proposed. We deal with several ontologies. A specific ontology is created for every data source and it corresponds to the data logical structure. Then all base ontologies are generalized to a domain ontology that serves the purpose of semantic data model against which all end user applications are implemented.

II. General Architecture

Fig. 1 depicts the general architecture for semantic querying of heterogeneous data sources. At the bottom is a data source layer where all external heterogeneous data sources are represented. They provide data to the data mediation layer with the help of base ontologies and semantic lifting. The data mediation layer uniforms different base ontologies to a single semantic model that is exported for use by end-user applications. Above all is the application layer that represents all end-user applications created against the domain model. Application layer utilizes semantic repositories, reasoners and SPARQL [24] query language to build queries against the domain model and infer results.

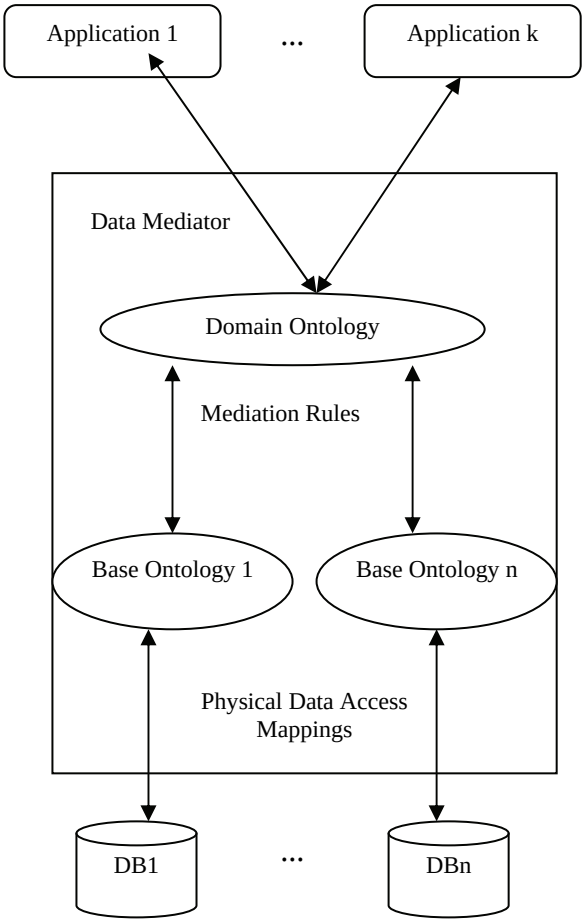

Fig. 1: General Architecture of semantic querying system.

III. Data Source Layer

Data source layer can be any data source that is accessible over the network and available to external applications. It can be standard relational databases, object databases, enterprise information systems, web feeds, web services, etc. For every data source a XML schema is created that describes logical

[1]Ivo Marinchev is with the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Bl. 2, Sofia 1311, Bulgaria, E-mail: ivo@iinf.bas.bg

*This work was partially supported by Research Project No. D-002-189 funded by the National Science Fund of Bulgaria

structure of the corresponding data. On top of this XML schema a base ontology is created that is a semantic description/explanation of the data source. The ontology is encoded in OWL [25]. Based on the XML schema and the ontology (semantic schema) a transformation rules are created that transforms between physical representations of the data and its form as ontology instances and vice versa. Transformation between the physical representations of the data and its form as ontology instance is named lifting in the semantic web literature. The opposite transformation is named lowering. Both are required during execution time for communication between semantic and non-semantic layers of the system as almost all data that exist today are not semantically annotated. Hence pure semantic practical system can not be build at the moment and in the near future and every such system will require some form of lifting.

One example for lifting data from the SINUS project [23] is given the fragment of XML instance file

```
<?xml version="1.0" encoding="utf-8" ?>
……
<Identification>
   <Location>
        <Area>Sofia</Area>
   </Location>
<Identification>
……
```

it is lifted to the following fragment of ontology instance

```
<owl:Thing rdf:about="#OWLClass_Province_65">
   <rdf:type rdf:resource="#OWLClass_Province" />
   <rdfs:label>Sofia</rdfs:label>
</owl:Thing>

<sinus:OWLClass_ObjectLocationAddress
   rdf:about="#OWLClass_ObjectLocationAddress35”>
   <rdf:type
      rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
   <sinus:OWLObjectProperty_has_Province
      rdf:resource="#OWLClass_Province_65" />
</ sinus:OWLClass_ObjectLocationAddress>
```

Note that on the semantic level the lifting produces two instances. One is the province of Sofia that is an instance of the ontology class OWLClass_Province and the other one is the instance of the class OWLClass_ObjectLocationAddress that has object property has_Province that refers to the province instance. On the semantic level we have two instances because the Sofia is lifted from String to the instance of ontology class. Hence it is no longer just sequence of characters but inherits all the semantics that stem from its corresponding class.

The last thing that is part of the physical data access mapping subsystem is the information about communication protocols. It is required so that the data mediation layer can communicate with corresponding data management software.

IV. Data Mediation Layer

Different data sources are expected to use different base ontologies for the annotation and interpretation of their data. Such differences hamper interoperation between applications and hamper reuse of data and ontologies across data bases. Reuse of data and interoperation between applications on the Semantic Web can be achieved by ontology merging, ontology mapping, and ontology alignment.

A. Ontology merging

In areas where ontologies do not overlap ontology merging can be implemented. As a side effect of ontology merging the newly created ontology can be shared between legacy applications which used the original ontologies. This ontology can now be used to enable interoperation between applications on the Semantic Web.

Ontology merging is the creation of a new ontology from two or more source ontologies. The new ontology will unify and in general replace the original source ontologies.

There are many different approaches to ontology merging found in the scientific literature. Some of them are [15], [20], [4] also different research tools are implemented PROMPT [17], OntoMerge [6], FCA-Merge [9] based on Formal concept analysis [4], OntoMorph [11].

B. Ontology mapping

In the case of ontology mappings, semantic overlap between ontologies needs to be detected and described using a formal language. Such a mapping can then be used for querying across ontologies, transforming data between representations, etc. The mappings are used to integrate autonomous heterogeneous applications over the Semantic Web.

An ontology mapping M is a (declarative) specification of the semantic overlap between two ontologies Os and Ot . This mapping can be one-way (injective) or two-way (bijective). In an injective mapping we specify how to express terms in Ot using terms from Os is a way that is not easily invertible. A bijective mapping works both ways, i.e. a term in Ot is expressed using terms of Os and the other way around [21].

Some tools the facilitate ontology mappings are MAFRA [1] and [19], OntoMap [3], RDFT [5].

One practical consideration related to mapping language is that it is better to be part of the ontology language itself or at least the widespread reasoners to be able to interpret it out of the box. Then any third part software as semantic repositories, reasoners, etc, can be reused without costly modifications.

C. Ontology matching/alignment

Ontology matching is the process of discovering similarities between two source ontologies. The result of a matching operation is a specification of similarities between two ontologies. Ontology matching is done through application of the Match operator [7].

Some approaches to ontology alignment are described in AnchorPROMPT [17], [18] and GLUE [2], Semantic Matching [8], QOM -Quick Ontology Mapping [13] and [14]

In our architecture we need a special case of ontology mapping, merging and alignment where the source ontologies remain, alongside of mappings to the domain ontology. In this case, the source ontologies can maintain their instance stores.

All the semantic information on this layer of the system is stored in some semantic repository as OWLIM [16], Sesame [22], or Jena [12].

Semantic repositories are engines similar to other database management systems (DBMS). Their main function is to support efficient storage, querying, and management of formal knowledge and semantically annotated data. The main functionalities of semantic repositories that distinguished them from other data management systems are:

- use ontologies as semantic schemata, which allows them to automatically reason about the data;
- work with generic physical data models, which allow them to easily adopt updates and extensions to the schemata, i.e. in the structure of the data;
- can be described as RDF-based column stores with inference capabilities.

V. Application Layer

This layer consists of all end user applications created against the domain model. They access the data with the help of SPARQL [24] query language against the used semantic repository. As mentioned above semantic repositories support inference capabilities. Thus using the semantics of the schemata/ontologies, semantic repositories can infer implicit information and return it during query evaluation. To illustrate the benefit of automated interpretation (or reasoning), consider a query about telecom companies in Europe: If an ontology describes the nesting of industry sectors and geographical areas, then a semantic repository would know to return mobile operators in the UK even though it has not been explicitly told that any particular UK mobile operator is also a European telecoms company.

VI. Querying Heterogeneous Data Sources

Mediation between ontologies is established in order to solve a particular problem in interoperability between ontologies. The most important use case for ontology mediation throughout this paper is querying.

Mediation between ontologies enables querying of one ontology in terms of another. This type of querying needs to be supported by the mediation component.

Such querying can be achieved in two principled ways: (1) by loading the source and target ontologies, together with the mapping rules, in the reasoner and then posing queries and (2) by rewriting queries in terms of the target ontology to queries in terms of the source ontology and then querying the source knowledge base, after which the query answers must be transformed to the target ontology.

Both ways have advantages and disadvantages. In case all ontologies along with the mapping rules are loaded in the reasoner, one can pose simple queries and immediately retrieve the answers in terms of the target ontology. The additional steps of rewriting the query and transforming the answers are not required. Disadvantage is that the reasoner must have access to the instance store which corresponds with the source ontology. Such an instance store would typically be a relational database and thus the reasoner must be aware how to translate queries on the ontology concepts to queries in the relational database and have access to the database to execute the queries.

In the second case, the additional steps of query rewriting and transformation of the query results are required. Especially query rewriting is a very complicated and costly task. This scenario is appropriate in case the source knowledge base exposes only a simple query interface and there is no access to the instance store.

VII. Conclusion

In this paper we presented general architecture for semantic querying of heterogeneous data sources. Important part of every implementation of this architecture is ontology mediation. We enumerated several approaches to ontology mediation that can generally be classified in three groups – ontology merging, ontology mapping, and ontology alignment. Our architecture require a special case of ontology mapping, merging and alignment where the source ontologies remain, alongside of mappings to the domain ontology. In this case, the source ontologies can maintain their instance stores.

References

[1] Alexander Maedche, Boris Motik, Nu no Silva, and Raphael Volz. *Mafra a mapping framework for distributed ontologies.* In Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW-2002, Madrid, Spain, 2002.

[2] AnHai Doan, Jazant Madhaven, Pedro Domingos, and Alon Halevy. *Ontology matching: A machine learning approach.* In Ste_en Staab and Rudi Studer, editors, Handbook on Ontologies in Information Systems, pages 397 - 416. Springer-Verlag, 2004.

[3] Atanas Kiryakov, Kiril Iv. Simov, and Marin Dimitrov. Ontomap: *The upper-ontology portal.* In Proceedings of "Formal Ontology in Information Systems", Ogunquit, Maine, 2001.

[4] Bernhard Ganter and Rudolph Wille. *Formal concept analysis: Mathematical Foundations.* Springer, Berlin-Heidelberg, 1999.

[5] Borys Omelayenko and Dieter Fensel. *A two-layered integration approach for product information in B2B e-commerce.* In Proceedings of the Second Intenational Conference on Electronic Commerce and Web Technologies (EC WEB-2001), Munich, Germany, 2001. Springer-Verlag.

[6] Dejing Dou, Drew McDermott, and Peishen Qi. *Ontology translation by ontology merging and automated reasoning.* In Proc. EKAW2002 Workshop on Ontologies for Multi-Agent Systems, pages 3 - 18, 2002.

[7] Erhard Rahm and Philip A. Bernstein. *A survey of approaches to automatic schema matching.* VLDB Journal: Very Large Data Bases, 10(4):334 - 350, 2001.
[8] Fausto Giunchiglia and Pavel Shvaiko. *Semantic matching.* The Knowledge Engineering Review, 18(3):265 - 280, 2004.
[9] Gerd Stumme and Alexander Maedche. *Fca-merge: Bottom-up merging of ontologies.* In 7th Intl. Conf. on Artificial Intelligence (IJCAI '01), pages 225 - 230, Seattle, WA, USA, 2001.
[10] A. Y. Halevy, *Answering queries using views: a survey.* In: VLDB Journal. Vol.10, No.4, pp. 270-294, 2001.
[11] Hans Chalupsky. OntoMorph: *A translation system for symbolic knowledge.* In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR 2000, Principles of Knowledge Representation and Reasoning Proceedings of the Seventh International Conference, pages 471 - 482, Breckenridge, Colorado, USA, 2000. Morgan Kaufmann Publishers.
[12] Jena http://jena.sourceforge.net/
[13] Marc Ehrig and Steffen Staab. *QOM - quick ontology mapping.* In Frank van Harmelen, Sheila McIlraith, and Dimitris Plexousakis, editors, Proceedings of the Third International Semantic Web Conference (ISWC2004), LNCS, pages 683 - 696, Hiroshima, Japan, 2004. Springer.
[14] Marc Ehrig and York Sure. *Ontology mapping - an integrated approach.* In Proceedings of the First European Semantic Web Symposium, ESWS 2004, volume 3053 of Lecture Notes in Computer Science, pages 76 - 91, Heraklion, Greece, May 2004. Springer Verlag.
[15] Michel Klein. *Combining and relating ontologies: an analysis of problems and solutions.* In Asuncion Gomez-Perez, Michael Gruninger, Heiner Stuckenschmidt, and Michael Uschold, editors, Workshop on Ontologies and Information Sharing, IJCAI'01, Seattle, USA, August 4 - 5, 2001.
[16] OWLIM http://www.ontotext.com/owlim/
[17] Natalya F. Noy and Mark A. Musen. *Anchor-prompt: Using non-local context for semantic matching*. In Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, USA, 2000.
[18] Natalya F. Noy and Mark A. Musen. *Prompt: Algorithm and tool for automated ontology merging and alignment.* In Proc. 17th Natl. Conf. On Artificial Intelligence (AAAI2000), Austin, Texas, USA, July/August 2000.
[19] Nuno Silva and Jo ao Rocha. *Service-oriented ontology mapping system.* In Proceedings of the Workshop on Semantic Integration of the International Semantic Web Conference (ISWC2003), Sanibel Island, USA, 2003.
[20] Pepijn R. S. Visser, Dean M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. *An analysis of ontological mismatches: Heterogeneity versus interoperability*. In AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA, 1997.
[21] Scharffe F. and Bruijn J.D., *A language to specify mappings between ontologies* In Proceedings of SITIS, 2005, pp.267-271.
[22] Sesame http://www.openrdf.org/
[23] SINUS project http://sinus.iinf.bas.bg/
[24] SparcQL query language http://www.w3.org/TR/rdf-sparql-query/
[25] Web Ontology Language (OWL) http://www.w3.org/TR/owl2-overview/